

---

# Rapport de refonte d'une application de jeu vidéo

---

ARIANE ZHANG  
XUE YANG

UE CPA  
MASTER INFORMATIQUE

*Tuteur(s) université :*  
BINH-MINH BUI-XUAN  
KATIA AMICHI  
ALFRED DEIVASSAGAYAME  
ARTHUR ESCRIOU

mars 2024

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Objectif du jeu</b>	<b>2</b>
2.1	But du jeu . . . . .	2
2.2	Règles du jeu . . . . .	2
<b>3</b>	<b>Analyse Fonctionnelle</b>	<b>3</b>
3.1	Histoires d'utilisateurs . . . . .	3
3.2	Cas d'utilisation . . . . .	3
<b>4</b>	<b>Plan de Développement</b>	<b>4</b>
4.1	Recherche . . . . .	4
4.2	Conception . . . . .	5
4.3	Produit . . . . .	5
4.4	Développement . . . . .	5
4.5	Rapport et Présentation . . . . .	5
<b>5</b>	<b>Architecture Générale</b>	<b>6</b>
5.1	Couche Client . . . . .	6
5.2	Couche Data . . . . .	7
5.3	Couche Serveur . . . . .	7
5.3.1	Trajectoires des Ennemis . . . . .	7
5.3.2	Logique de Collision . . . . .	8
5.4	Conclusion Technique . . . . .	9
<b>6</b>	<b>Résultat</b>	<b>9</b>
<b>7</b>	<b>Conclusion et Retours d'Expérience</b>	<b>11</b>
	<b>Références</b>	<b>12</b>

# 1 Introduction

Dans le cadre du cours CPA, notre équipe s'est engagée à développer un jeu vidéo de plateforme 2D intitulé *Super Dog*. Les membres de notre équipe, Ariane ZHANG et Xue YANG, ont été responsables de la définition du concept et des objectifs du jeu. Pour garantir une gestion de projet systématique et efficace, nous avons utilisé un diagramme de Gantt pour planifier, exécuter et surveiller chaque étape du développement.

L'objectif de ce projet est non seulement de rendre hommage aux jeux de plateforme 2D classiques, mais aussi d'apprendre et d'appliquer les techniques fondamentales du développement web moderne afin de créer une expérience de jeu qui combine des éléments nostalgiques avec un design innovant. Parmi les nombreuses technologies modernes disponibles, nous avons choisi HTML5, CSS3 et JavaScript comme outils de développement. L'élément `<canvas>` de HTML5 nous offre une base solide pour le dessin graphique, tandis que CSS3 permet d'enrichir le jeu avec un style visuel attrayant et des animations fluides. JavaScript vanille, apprécié pour sa simplicité sans dépendances externes et sa compatibilité étendue avec tous les navigateurs modernes, garantit une exécution optimale du jeu.

Inspirés par les tutoriels de Frank's Laboratory[1], nous avons vu comment les technologies de base peuvent être utilisées pour créer des jeux à la fois amusants et fonctionnels. Cette approche d'apprentissage par la pratique nous a permis de comprendre profondément les mécanismes internes des jeux et de renforcer des compétences essentielles telles que la gestion d'état, la logique de détection des collisions et la conception d'animations image par image. Engagés à transformer la théorie en pratique, nous avons appris à écrire un code propre, à gérer des interactions utilisateur complexes et à optimiser les performances des jeux web.

En participant au projet *Super Dog*, chaque membre de l'équipe a acquis une expérience précieuse, jetant les bases pour des projets de programmation plus complexes dans l'avenir.

## 2 Objectif du jeu

### 2.1 But du jeu

Le concept central de notre jeu *Super Dog* implique le joueur qui contrôle un chien aux super-pouvoirs, capable de tuer des monstres ou de se faire tuer par eux. À chaque monstre vaincu, Super Dog gagne 1 point. En revanche, si Super Dog est vaincu par un monstre, il perd une vie. Le jeu est minuté : le joueur dispose de 50 secondes pour accumuler les points. La partie est perdue si le joueur épuise toutes ses vies ou si son score est inférieur à 20 points quand le chrono arrive à terme. L'objectif est donc de survivre pendant les 50 secondes allouées tout en éliminant le maximum de monstres.

### 2.2 Règles du jeu

Voici les commandes permettant d'interagir avec *Super Dog* :

- **Sauter** : La flèche vers le haut permet au personnage de sauter.
- **Plonger** : La flèche vers le bas est utilisée pour plonger.
- **Courir** : Pour courir, le joueur doit utiliser les flèches de gauche ou de droite.
- **Rouler** : La touche Entrée est assignée à l'action de rouler.
- Les monstres ne peuvent être éliminés que par un roulé ou un plongeon.

- Un contact avec un ennemi sans l’usage de ces techniques entraîne la perte d’une vie.
- Le joueur débute la partie avec trois vies.
- Pour gagner, il faut marquer plus de 20 points avant la fin du décompte des 50 secondes.

La maîtrise de ces commandes est cruciale pour permettre aux joueurs de naviguer habilement dans le jeu et d’optimiser leurs chances de victoire.

## 3 Analyse Fonctionnelle

Cette section présente les fonctionnalités principales du jeu *Super Dog* à travers les histoires d’utilisateurs et les cas d’utilisation, définissant ainsi les interactions attendues entre le joueur et le jeu.

### 3.1 Histoires d’utilisateurs

Les histoires d’utilisateurs suivantes ont été identifiées pour guider le développement des fonctionnalités du jeu :

1. En tant que joueur, je souhaite que les commandes du jeu soient intuitives et faciles à maîtriser pour pouvoir rapidement m’immerger dans l’expérience de jeu.
2. En tant que joueur, je désire rencontrer différents niveaux de difficulté avec les ennemis afin d’augmenter le challenge et le plaisir du jeu.
3. En tant que joueur, je veux obtenir des points en éliminant les ennemis et en accomplissant des tâches pour me sentir motivé à progresser et à battre mon meilleur score.
4. En tant que joueur, je préfère avoir plusieurs vies dans le jeu pour me donner la chance de corriger les erreurs occasionnelles.
5. En tant que joueur, j’apprécie que le jeu soit chronométré pour augmenter la tension et le défi du gameplay.
6. En tant que joueur, je veux des musiques de fond et des effets sonores adaptés qui rendent le jeu plus captivant et agréable à jouer.

### 3.2 Cas d’utilisation

Les cas d’utilisation suivants décrivent les actions de base que le joueur peut effectuer dans le jeu :

1. Cas d’utilisation : **Courir**
  - Action du joueur : Appuyer sur la touche flèche gauche ou droite.
  - Réponse du jeu : Super Dog court vers la gauche ou la droite.
  - Résultat : Évite ou entre en collision avec un ennemi.
2. Cas d’utilisation : **Sauter**
  - Action du joueur : Appuyer sur la touche flèche vers le haut.
  - Réponse du jeu : Super Dog saute pour éviter ou entrer en collision avec un ennemi.
  - Résultat : Évite avec succès ou vainc un ennemi.
3. Cas d’utilisation : **Attaque roulée**
  - Action du joueur : Appuyer sur la touche Entrée.

- Réponse du jeu : Super Dog effectue une attaque roulée.
  - Résultat : Vainc les ennemis et obtient des points.
4. Cas d'utilisation : **S'asseoir**
- Action du joueur : Appuyer sur la touche flèche vers le bas.
  - Réponse du jeu : Super Dog s'assied lorsqu'il est au sol.
  - Résultat : Fait pause à la position actuelle dans le jeu.
5. Cas d'utilisation : **Plongeon**
- Action du joueur : Appuyer sur la touche flèche vers le bas.
  - Réponse du jeu : Super Dog plonge à grande vitesse vers le bas lorsqu'il est en l'air.
  - Résultat : Attaque et élimine les ennemis au contact.



FIGURE 1 – Wireframe de l'écran d'accueil du jeu

## 4 Plan de Développement

Notre projet de jeu *Super Dog* a été structuré en plusieurs phases clés, chaque phase étant conçue pour optimiser la gestion de notre temps et de nos ressources.

### 4.1 Recherche

La phase de recherche a été consacrée à l'établissement des fondations du projet. Durant cette période, nous avons :

- Effectué une analyse approfondie des jeux classiques en 2D pour sélectionner un concept qui rencontre à la fois l'intérêt de notre équipe et les exigences du cours.
- Conçu un diagramme de Gantt pour planifier notre projet de manière structurée.

## 4.2 Conception

La phase de conception s'est concentrée sur la création d'un cadre visuel et fonctionnel pour *Super Dog*, incluant :

- Le développement des wireframes UX/UI et l'utilisation d'outils tels que Figma pour le design préliminaire.
- La rédaction d'un document de conception de jeu détaillant le gameplay, l'histoire, les personnages et la conception des niveaux.

## 4.3 Produit

Cette étape a été axée sur la transformation des idées en un prototype fonctionnel, comprenant :

- La définition précise des spécifications fonctionnelles.
- Le développement et l'itération sur un prototype interactif initial du jeu.

## 4.4 Développement

Le cœur du développement s'est déroulé ici, où nous avons programmé et donné vie au jeu. Les tâches comprenaient :

- La construction de l'infrastructure technique nécessaire.
- La programmation des fonctionnalités du jeu et l'intégration complète du contenu.

## 4.5 Rapport et Présentation

La finalisation du projet se caractérise par la rédaction d'un rapport complet et la préparation d'une présentation du produit fini. Cette dernière phase inclut :

- L'élaboration d'une introduction et d'une analyse fonctionnelle reflétant le processus de développement dans son intégralité.
- La préparation de la partie technique, incluant l'architecture générale de l'application ainsi que les couches client, données et serveur.
- La conclusion et la rétrospective pour dresser le bilan et recueillir les enseignements.
- La présentation finale du jeu via une vidéo pitch pour partager notre travail accompli.

Une organisation méthodique des tâches et une gestion efficace du temps ont été cruciales pour respecter le calendrier établi et atteindre les objectifs définis, avec une approche flexible permettant des ajustements itératifs en fonction des retours obtenus à chaque étape du développement.

	Week 1 (15-19/03)	Week 2 (20-26/03)	Week 3 (27/03-02/04)	Week 4 (03-09/04)	Week 5 (10-16/04)	Week 6 (17-23/04)	Week 7 (24-30/04)	Week 8 (01-05/05)
Recherche	Réunion de lancement Membres de l'équipe : ZHANG Ariane, YANG Xue Jeu vidéo : Super Dog							
Conception		Wireframe UI/UX Figma pour le design préliminaire						
			Document de conception de jeu Définir le gameplay, l'histoire, les personnages, et la conception des niveaux					
Analyse fonctionnelle		Spécifications fonctionnelles Analyse fonctionnelle (usecase et/ou userstory)						
			Développement du prototype Construire un prototype interactif initial du jeu					
Développement			Construction de l'infrastructure Mettre en place l'environnement de développement selon la recherche technologique					
				Développement du jeu Programmer les fonctionnalités du jeu				
Rapport		Introduction et Analyse Fonctionnelle		Partie Technique Architecture générale de l'application et les couches client, data, serveur			Conclusion et Rétrospective	
Présentation					Écrire des scripts vidéo	Filmer des séquences vidéo	Organisez les clips et complétez la présentation	

FIGURE 2 – Plan de Développement

## 5 Architecture Générale

*Super Dog* est un jeu 2D qui fonctionne entièrement sur le client, utilisant l'élément `<canvas>` de HTML5 pour le rendu graphique et CSS3 pour définir le style des éléments de l'interface. La logique principale du jeu, le traitement des animations, les interactions des joueurs et la gestion de l'état du jeu sont tous réalisés via JavaScript.

### 5.1 Couche Client

La couche client est construite avec les fichiers principaux suivants :

- **index.html** : Fournit la structure HTML du jeu, y compris le canevas du jeu et la liste des commandes pour l'interaction du joueur.
- **style.css** : Définit le style visuel des éléments du jeu, y compris les polices, les couleurs et le positionnement des éléments.
- **background.js** : Gère le défilement des arrière-plans multicouches pour créer un effet de profondeur visuelle.
- **UI.js** : Gère l'affichage des scores, du temps, des vies et de l'interface de fin de jeu.
- **main.js** : Point d'entrée de la logique du jeu, gérant l'initialisation de la boucle de jeu, l'écoute des événements et la mise à jour et le rendu des objets du jeu.

`main.js` importe des modules tels que `Joueur`, `Background`, `EnnemiVolant`, `EnnemiTerrain`, `EnnemiGrimpant` et `UI`. Ces classes modulaires gèrent des fonctions spécifiques du jeu telles que le contrôle des personnages, le rendu d'arrière-plan et la gestion de l'interface utilisateur.

## 5.2 Couche Data

Le jeu n'utilise pas de base de données côté serveur, toutes les gestions des données sont effectuées via des variables et des objets dans JavaScript. Par exemple, les scores et les vies des joueurs sont mis à jour dans la fonction de détection de collision dans `joueur.js` et sont affichés en temps réel dans l'UI. La génération et la suppression des ennemis sont gérées par des tableaux stockant et éliminant différents types d'ennemis.

## 5.3 Couche Serveur

Dans *Super Dog*, la "couche serveur" fait référence à la partie qui traite la logique du jeu.

- **joueur.js** et **joueurEtats.js** : Gèrent les états du joueur et la logique de transition entre ces états, y compris l'assise, la course, le saut, la chute, le roulé, la plongée et l'impact.
- **ennemi.js** : Définit les types d'ennemis et contrôle leur génération et leur comportement.
- **collisionAnimation.js** : Génère les animations correspondantes lors des collisions, telles que les explosions ou les éclaboussures.

La classe Joueur contient la logique de contrôle du joueur, l'algorithme de détection de collision et la gestion des changements d'état en fonction des entrées du clavier.

### 5.3.1 Trajectoires des Ennemis

Dans *Super Dog*, les trajectoires et les animations des ennemis sont soigneusement conçues pour ajouter de la variété et du défi à l'expérience de jeu. Différents types d'ennemis utilisent des algorithmes de mouvement spécifiques :

- Les ennemis volants se déplacent horizontalement à différentes vitesses ( $Math.random() + 1$ ) ; et suivent un mouvement vertical sinusoïdal. Cela est réalisé en changeant la position verticale en fonction de la valeur sinusoïdale de l'angle qui augmente à chaque mise à jour du jeu ( $this.y += Math.sin(this.angle)$ ). Le changement d'angle ( $this.va$ ) ajoute un élément aléatoire aux trajectoires, rendant le mouvement de chaque ennemi volant unique.
- Les ennemis terrestres restent au sol, et le joueur doit calculer leur mouvement pour attaquer ou esquiver.
- Les ennemis grimpants apparaissent de manière aléatoire et se déplacent verticalement. Lorsqu'ils atteignent le sol ou un certain point haut, ils peuvent changer de direction, contrôlés par la vérification des limites et la réversibilité de la vitesse verticale ( $this.speedY$ ).



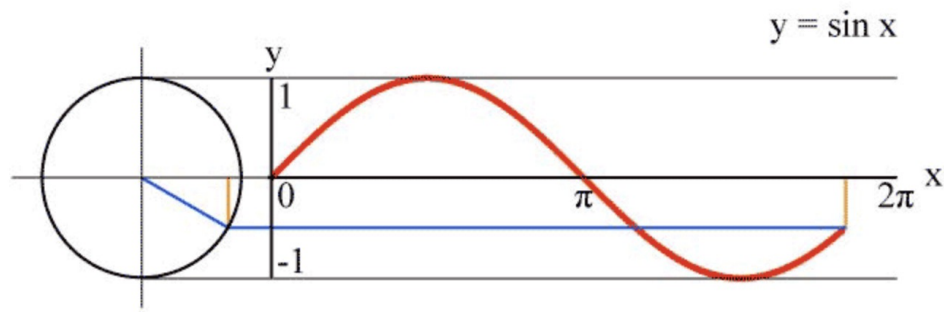


FIGURE 3 – Illustration du mouvement vertical sinusoïdal et de la variation de la vitesse angulaire des ennemis

### 5.3.2 Logique de Collision

La détection de collision dans *Super Dog* est un élément clé du gameplay, affectant directement le score et la survie du joueur. La classe Joueur utilise une série de vérifications pour détecter les intersections avec les ennemis :

- Cette logique s'appuie sur les coordonnées des axes  $X$  et  $Y$  ainsi que sur les dimensions du sprite pour établir la zone de collision.
- Lorsqu'une collision est détectée, le jeu déclenche des effets sonores et des animations spécifiques de la classe CollisionAnimation, qui affichent des effets visuels tels que des explosions ou des éclatements pour signaler la collision.
- De plus, la collision affecte l'état du joueur - si le joueur est en état d'attaque (roulant ou plongeant), il éliminera les ennemis ; sinon, le joueur perd une vie, et l'état du joueur passe à "frappé" avec une animation de frustration ou de réaction.
- Une fois que les ennemis sortent de l'écran ou après une collision, ils sont marqués pour suppression (*markedForDeletion*), les éliminant du jeu et empêchant tout traitement ultérieur.

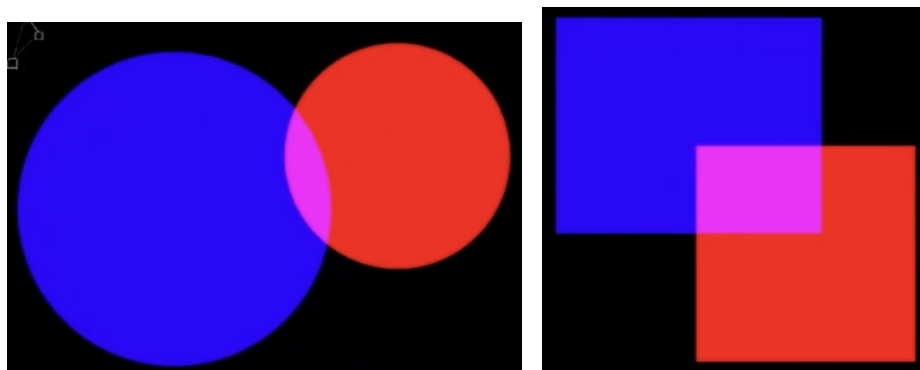


FIGURE 4 – Représentation schématique des zones de collision

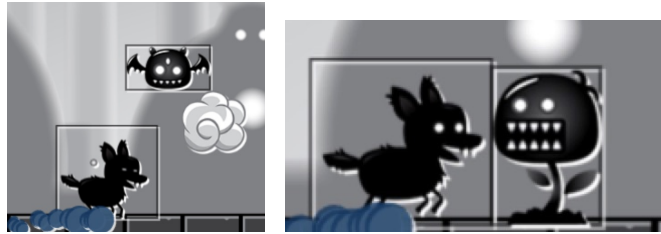


FIGURE 5 – Représentation des animations de collision

Cette gestion précise des trajectoires et des collisions contribue à l'aspect stratégique du jeu, forçant les joueurs à apprendre et à s'adapter aux différents modèles de mouvement des ennemis pour progresser et obtenir des scores élevés.

## 5.4 Conclusion Technique

Tous les éléments du jeu sont définis à travers des classes JavaScript modulaires, assurant une structure de code claire et facile à maintenir. Le jeu s'appuie sur HTML5 et CSS3 pour fournir une interface interactive riche, et JavaScript pour réaliser des animations fluides, une détection précise des collisions et la gestion de l'état du jeu, garantissant une expérience de jeu performante et agréable.

# 6 Résultat

1. L'écran d'accueil du jeu et les instructions de jeu.



FIGURE 6 – L'écran d'accueil du jeu et les instructions de jeu

2. Modes d'attaque



FIGURE 7 – Schéma des modes d'attaque

### 3. Page de fin de jeu



FIGURE 8 – Interface de victoire du joueur



FIGURE 9 – Interface de défaite du joueur

## 7 Conclusion et Retours d'Expérience

### 1. Synthèse du Projet

*Super Dog* représente un projet intégrant diverses compétences, qui améliore et innove sur les jeux classiques en conjuguant développement web et divers algorithmes logiques. Dans ce projet, nous avons appris à concevoir et développer un jeu 2D interactif, qui est non seulement divertissant mais aussi renforce notre compréhension des mécanismes tels que la détection de collisions et les algorithmes de mouvement abordés en cours.

### 2. Difficultés Rencontrées

Tout au long du développement, nous avons rencontré plusieurs défis, notamment dans la mise en œuvre d'un mouvement fluide des sprites et la mise à jour des états. L'intégration de différents modules, tels que la détection des collisions, le rendu des animations et la gestion des états, a présenté des défis pour assurer une expérience de jeu cohérente.

### 3. Perspectives d'Amélioration

Bien que le jeu soit complet et agréable, il reste une marge d'amélioration. Sur la base des retours des utilisateurs, nous envisageons d'ajouter de nouveaux niveaux, types d'ennemis et éléments interactifs pour enrichir le gameplay. Nous explorons également l'utilisation de bases de données pour enregistrer les données historiques des joueurs, ainsi que l'introduction d'un système de comptes permettant aux joueurs de sauvegarder et de reprendre leur progression.

### 4. Conclusion

En somme, *Super Dog* a été un projet captivant pour notre équipe. Nous avons acquis de nombreuses compétences pratiques tout au long de ce projet, stimulant la fusion entre la technologie et la créativité. Nous sommes convaincus que les compétences et les connaissances acquises lors de cette aventure serviront de tremplin pour nos futures carrières dans le développement logiciel.

## Références

- [1] Franks laboratory. JavaScript Game Tutorial with HTML Canvas, April 2022.